

(12) UK Patent Application (19) GB (11) 2 240 201 (13) A
(43) Date of A publication 24.07.1991

(21) Application No 9027328.5

(22) Date of filing 18.12.1990

(30) Priority data

(31) 01327315

(32) 19.12.1989

(33) JP

(71) Applicant
Hitachi Ltd

(Incorporated in Japan)

6 Kanda Surugadai 4-chome, Chiyoda-ku, Tokyo,
Japan

(72) Inventor
Kelko Yamakawa

(74) Agent and/or Address for Service
Mewburn Ellis
2 Cursitor Street, London EC4A 1BQ, United Kingdom

(51) INT CL⁶
G06F 15/40

(52) UK CL (Edition K)
G4A AUD

(56) Documents cited
None

(58) Field of search
UK CL (Edition K) G4A AUD AUX
INT CL⁶ G06F

(54) Manipulating non-leaf entries in data tree

(57) A method for changing a tree structure of an OSI directory includes the steps of decoding a command for a non-leaf entry in the directory tree structure, issuing OSI directory standard commands in order in accordance with the decoding result, and executing, in response to the standard commands, operations corresponding to the standard commands for the non-leaf entry and its lower entries in the directory tree structure.

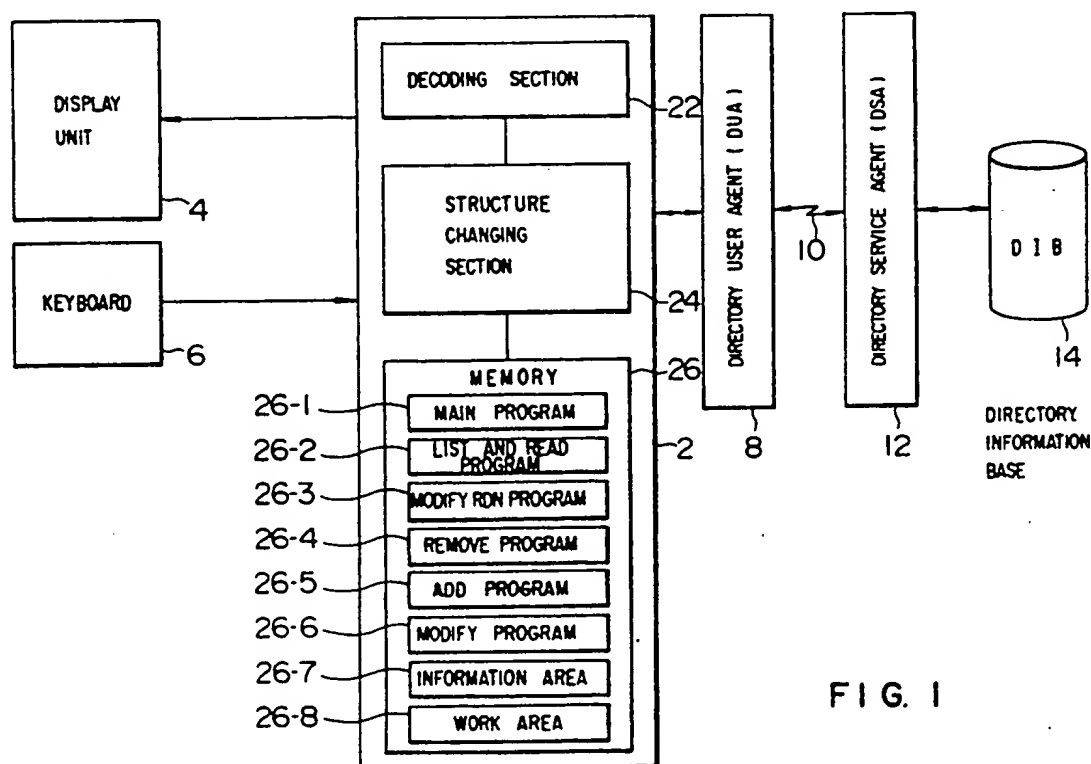


FIG. 1

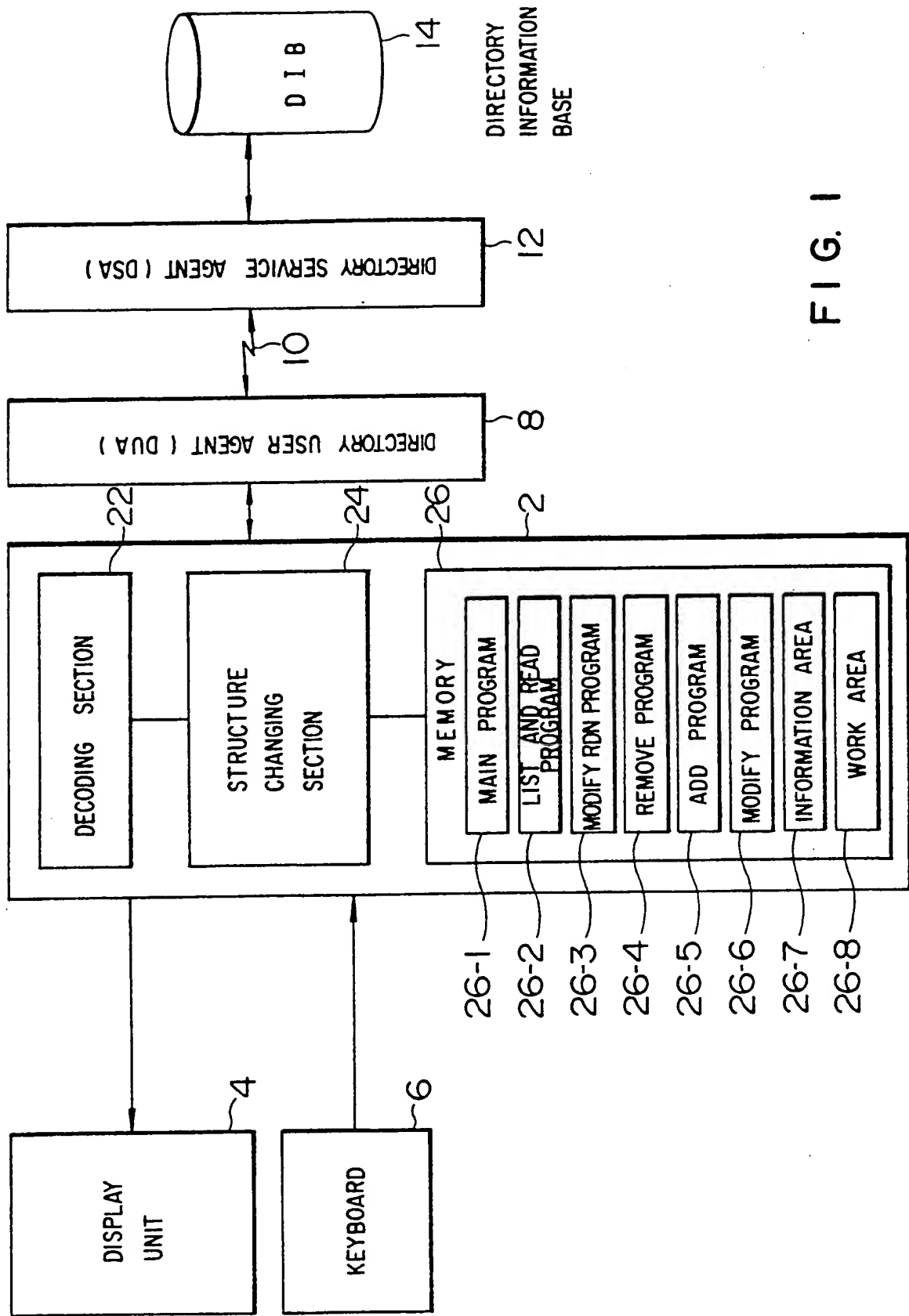


FIG. 2

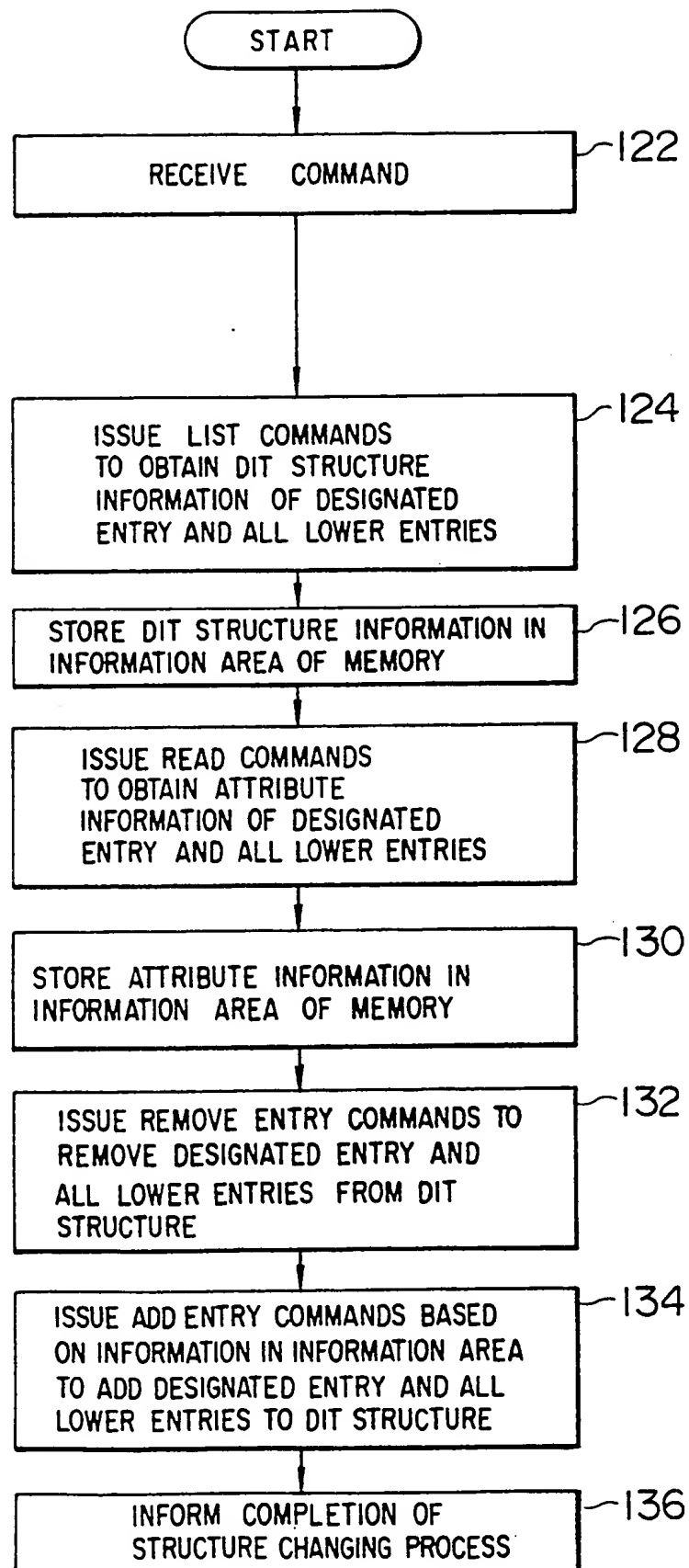


FIG. 3

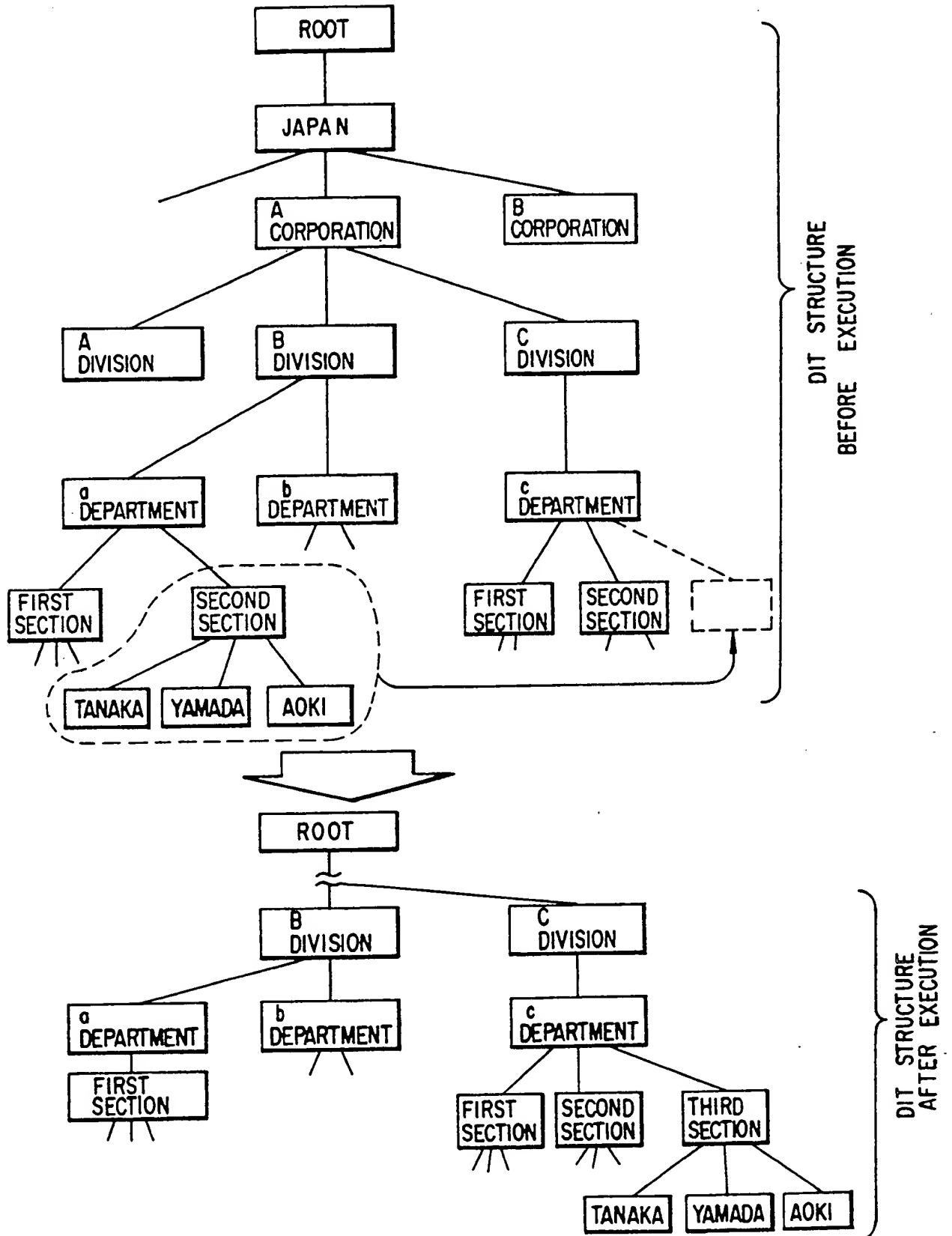


FIG. 4

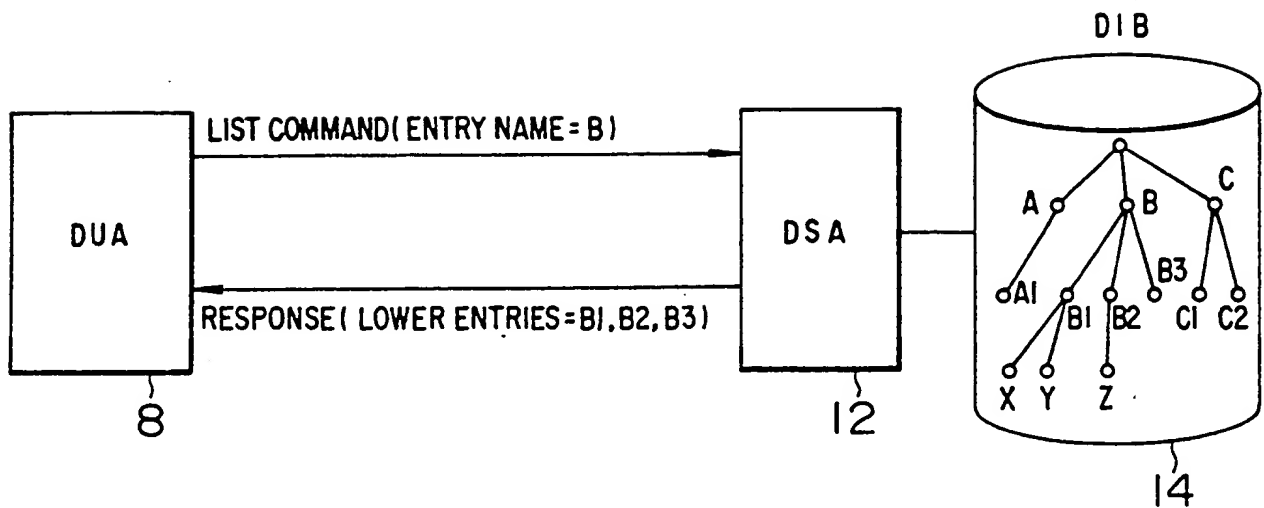


FIG. 5

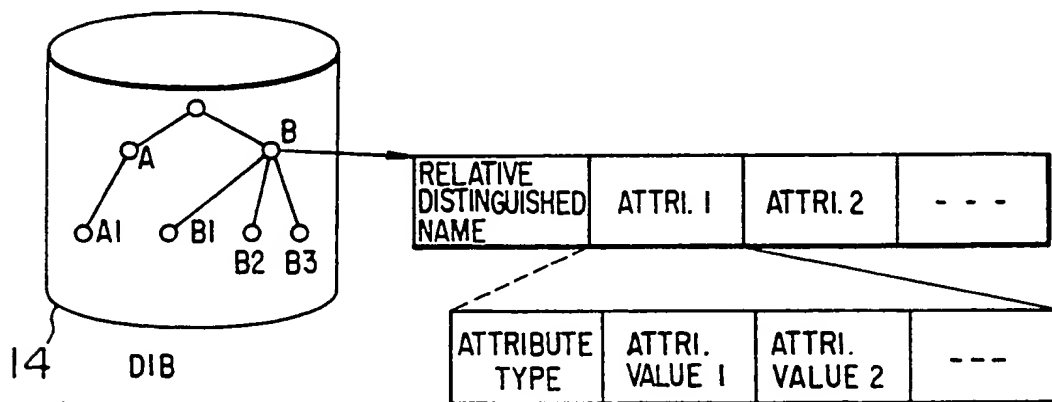


FIG. 6

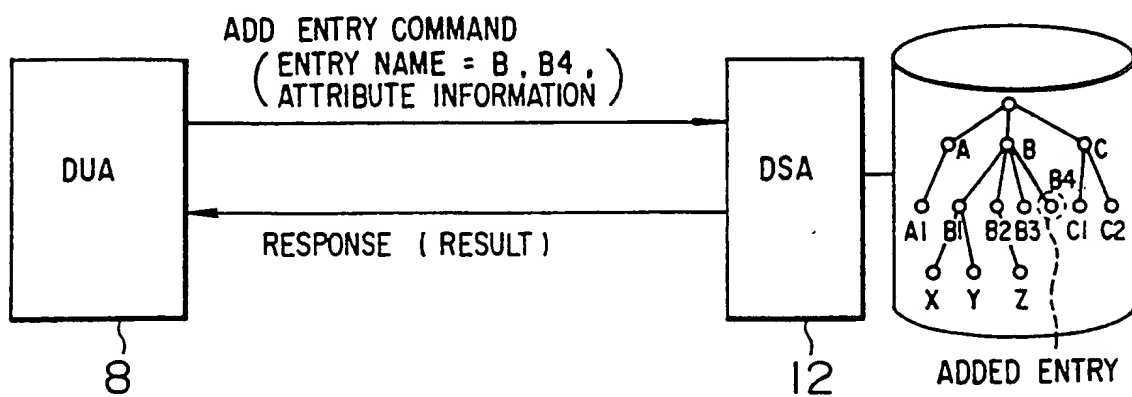
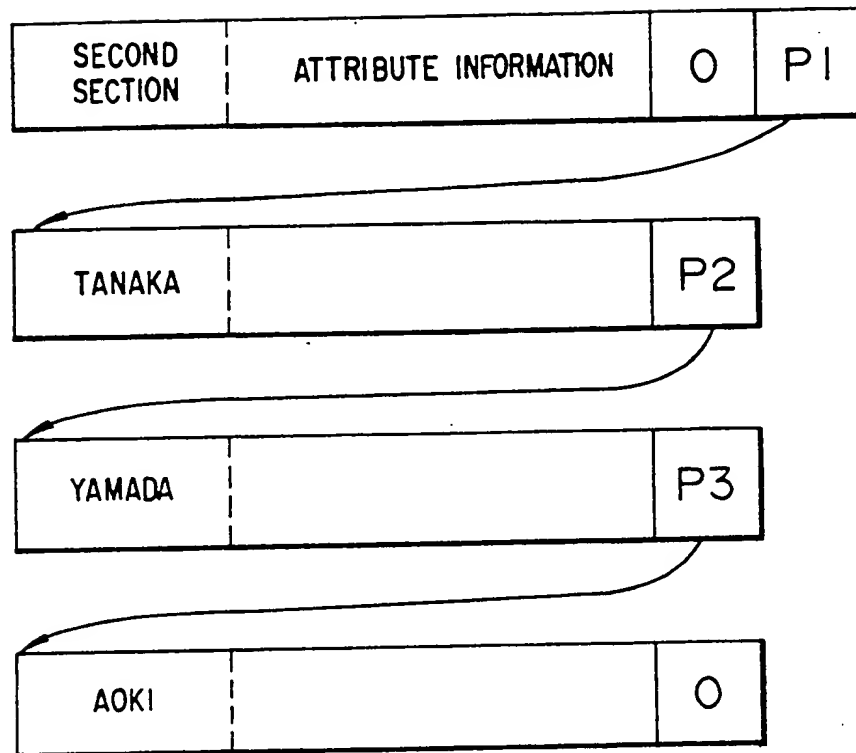


FIG. 7

26-7



POINTER FLAG

26-8

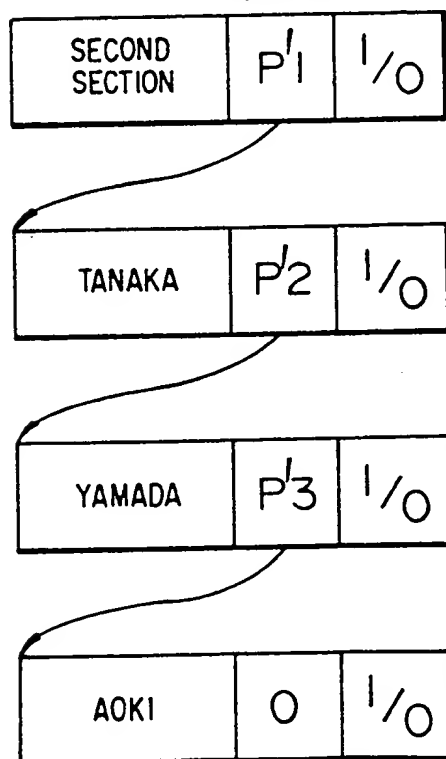


FIG. 8

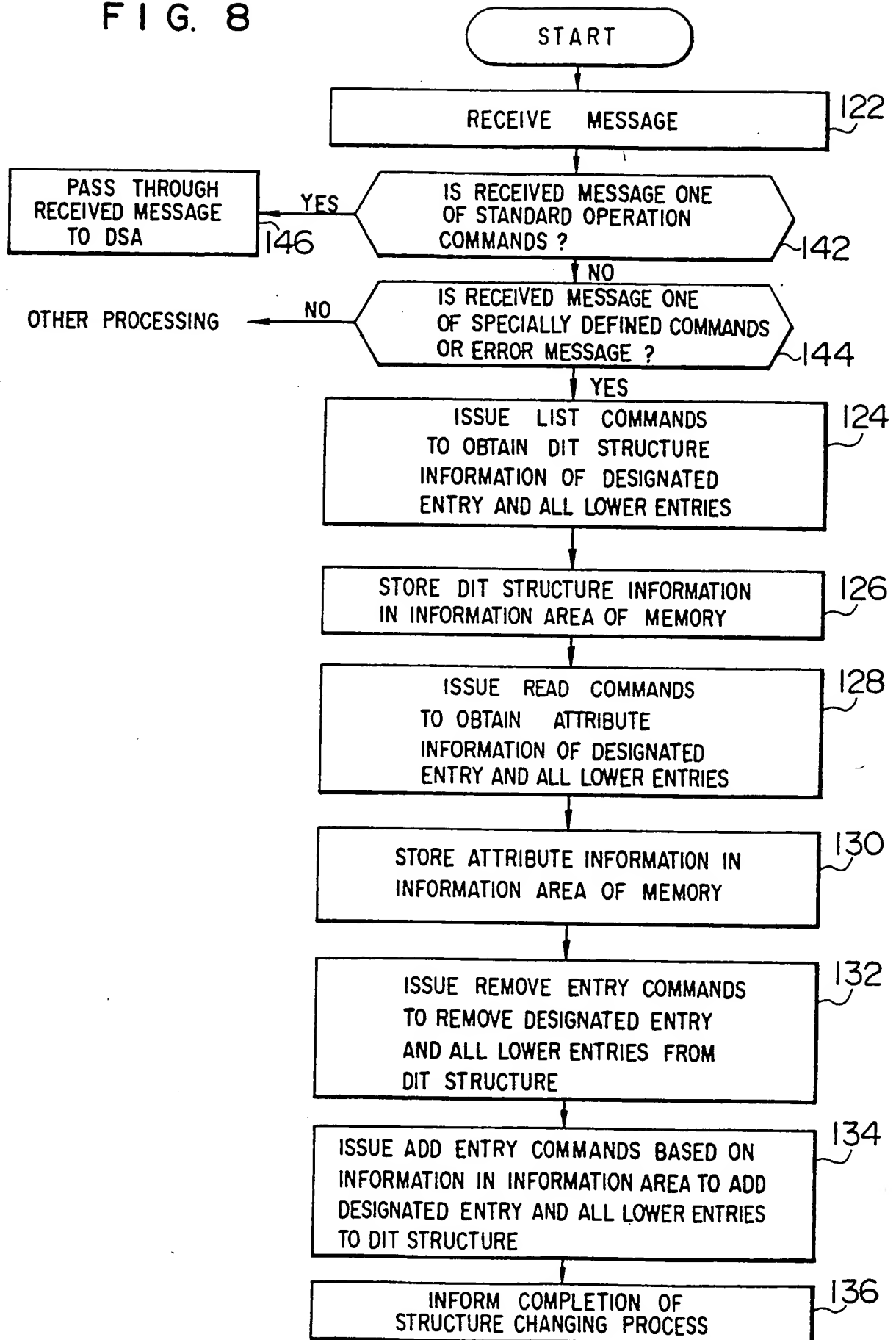
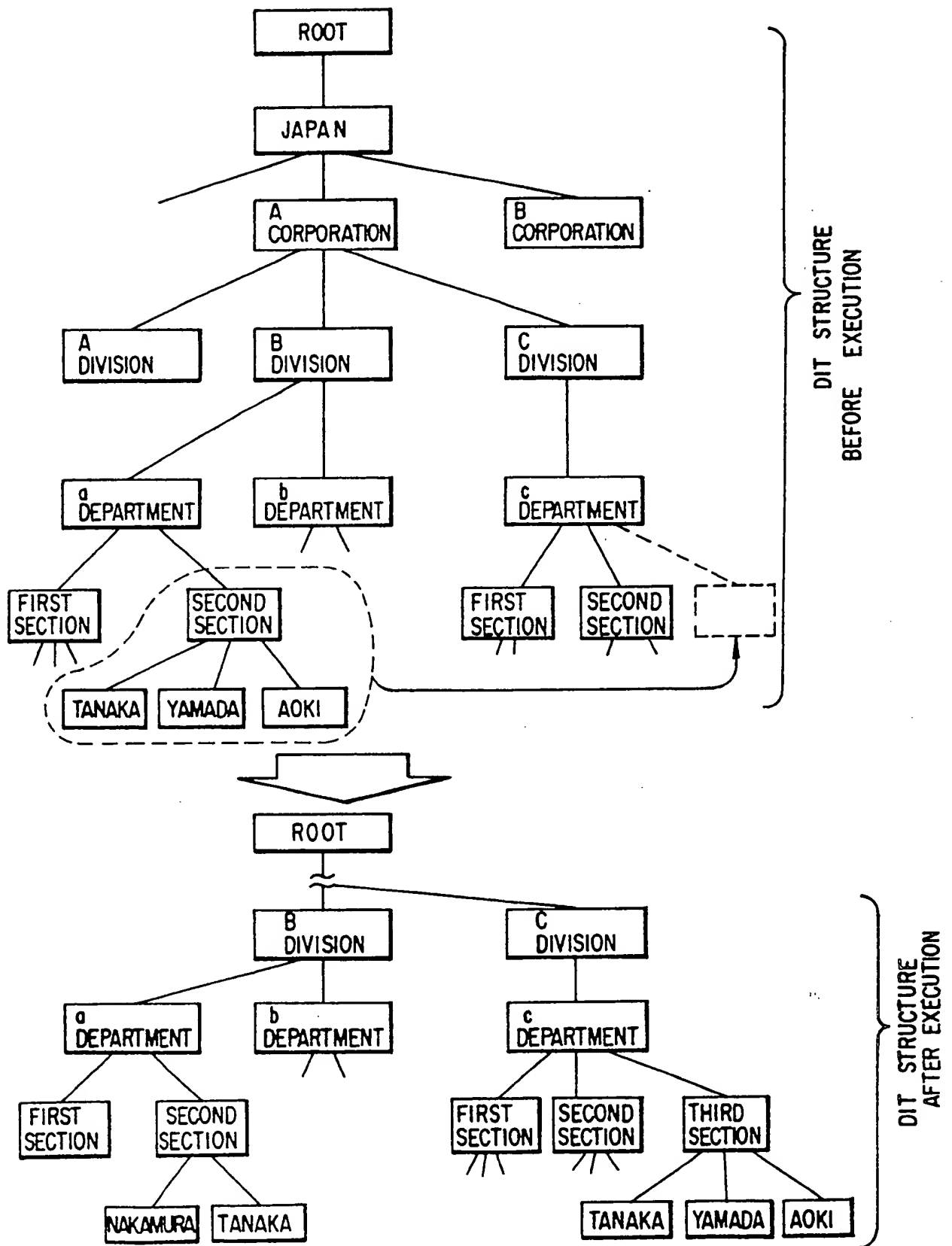


FIG. 9



METHOD FOR MANIPULATING NON-LEAF ENTRIES IN TREE
STRUCTURE OF OSI DIRECTORY INFORMATION AND DIRECTORY
SERVICE SYSTEM FOR THE SAME

1

The present invention relates to a system for
modifying non-leaf entries in OSI (Open System Intercon-
5 nection) directory of a tree structure, and more
particularly to a directory service system using an OSI
directory in which entries can be added, modified or
removed to or from a directory information tree (DIT).

10 Generally, in order to make it possible to
communicate with a destination terminal without knowing
physical arrangement of a network, in an OSI directory,
address information and attribute information of the
destination terminal are provided on the basis of a
15 logical name thereof, to be informed to a source
terminal.

A system for implementing such directory
service as mentioned above usually comprises a directory
user agent (DUA), a directory service agent (DSA) and a
20 directory information base (DIB). Ones to be managed by
the OSI director are called objects which are stored as
entries in the DIB. A structure in which these entries
are arranged in a tree form, based on correlation of
them with each other, is called a directory information

1 tree (DIT), the highest entry of this DIT is called a
root which is connected with lower entries. Of these
lower entries, entries which are not connected with
further lower entries are called leaf entries, and
5 entries which are not the leaf entries are called non-
leaf entries.

The commands which can be used in the above
system including the DUA and DSA, include LIST, READ,
ADD, MODIFY RDN, and so on. These commands, however,
10 operate for only leaf entries so that the system has a
drawback that non-leaf entries cannot be modified
through the commands. For example, in a case where a
certain section is shifted into another section or a
name of the section is changed, because of change in
15 organization without changing constituent elements of
the section, the inherent name of the section as a non-
leaf entry must be modified. In this case, the prior
art, however, cannot modify the inherent name by a
single operation or command.

20 As one of the prior arts related to the OSI
directory, there is "OSI - The Directory (ISO9594)"
proposed as Draft International Standard (DIS), and a
conventional technique related to a system based on this
DIS is disclosed in "Implementation of DIRECTORY (DSA)",
25 JOHOSHORI GAKKAI 38-th 5H-1/2/3. However, this
technique is mainly related to the directory (DSA).

A prior art for improving operatinality for a
directory user is disclosed in, for example, "Proposal

(
1 of Function for Supporting Directory User", JOHOSHORI
GAKKAI 35-th 5U-1. This prior art, however, relates to
search for directories and does not take into considera-
tion improvement of service for users such as addition,
5 modification, etc. of entries for DIT.

An object of the present invention is to solve
problems of the above prior art in a directory service
system and to provide an OSI directory service system
10 which can execute a request for modifying a non-leaf
entry in a directory tree structure through one
operation.

In accordance with the present invention, in
order to attain the above object, when the directory
15 service system receives the request with old and new
inherent names for modifying a inherent name of the non-
leaf entry, the standard commands are issued to the DSA
and the DUA so as to perform standard operation for the
DIB and the inherent name of the non-leaf entry is
20 modified. More specifically, the LIST commands are
issued to obtain the DIT structure information of a
designated entry having an old inherent name and its all
lower entries, the READ commands are issued to obtain
all attribute information of the designated entry and
25 the all lower entries, the REMOVE Entry commands are
issued to remove the designated entry and all the
entries from the DIT structure, and the ADD Entry

1 commands are issued to add the designated entry having
the new inherent name and all the lower entries to the
DIT structure, thus completing the modification of the
inherent name of the designated non-leaf entry.

5 The directory service system according to the
present invention comprises a memory enough to store
information of each of non-leaf entries. When the
directory service system receives the old and new
inherent name of the designated non-leaf entry from a
10 directory user, the old inherent name of the designated
non-leaf entry is modified to the new inherent name on
the basis of standard service (operations by the
commands of LIST, READ, REMOVE Entry and ADD Entry)
provided by the OSI directory, all lower entries than
15 the designated entry are added to the DIT structure as
the lower entries of the non-leaf entry with the new
inherent name, and completion of the DIT structure
modification process is informed to the user.

 In this way, the directory user has only to
20 issue one command to modify the inherent name of a
designated non-leaf entry and shift all the lower
entries of the designated non-leaf entry to the lower
side of the non-leaf entry with the new name.

 The present invention will now be described in
greater detail by way of example with reference to the
25 accompanying drawings wherein:-

 Fig. 1 is a block diagram showing arrangement of
a directory service system according to a first
embodiment of the present invention;

1 Fig. 2 is a flowchart for explaining an
operation of the first embodiment;

 Fig. 3 is a view for explaining modification
of a DIT structure;

5 Figs. 4, 5 and 6 are views for explaining a
LIST operation, a READ operation and an ADD operation
which are standard operations, respectively;

 Fig. 7 is a view for explaining data stored in
an information area of a work area in a memory;

10 Fig. 8 is a flowchart for explaining an
operation of a second embodiment of the present
invention; and

 Fig. 9 is a view for explaining modification
of the DIT structure in the second embodiment.

15

 Now, referring to accompanying drawings,
explanation will be given for a directory service system
for an OSI directory structure according to the present
invention.

20 Fig. 1 is a block diagram showing an arrange-
ment of a first embodiment of the directory service
system according to the present invention. Fig. 2 is a
flowchart for explaining an operation of the first
embodiment. Fig. 3 is a view for explaining modifica-
25 tion of a Directory Information Tree (DIT) structure.

 In Fig. 1, an interface unit 2 is a
characteristic portion of the present invention. A

1 display unit 4 displays output data from the interface
unit 2. When commands are input from a input device 6
such as a keyboard to interface unit 2, the interface
unit 2 outputs a train of commands. A Directory User
5 Agent (DUA) 8 is a standard user interface for utilizing
OSI directory service. The command train from the
interface unit 2 is sent to the DUA 8. A Director
Information Base (DIB) stores directory information in
the form of a tree structure. A Directory Service Agent
10 (DSA) 12 receives a command from the DUA 8 through a
communication network 10 to execute for the DIB 14 an
operation determined in accordance with the command, and
returns the executing result to the interface unit 2
through the network 10 and DUA 8. The display unit 4
15 displays that result sent from the interface unit 2.

Interface unit 2 includes a decoding section
22, a structure changing section 24 and a memory 26.
The section 22 decodes a command input from the keyboard
6. When the input command is one of predetermined
20 commands, i.e., the standard commands for the OSI
directory, the section 22 outputs the input commands to
the DUA 8. When the input command is not one of the
predetermined commands, the section 22 activates the
structure changing section 24 to execute a structure
25 changing processing. The memory 26 stores data and
programs necessary for the structure changing proces-
sing. Specifically, the memory 26 stores, in addition
to a main program 26-1 for controlling the DIT structure

1 changing processing through one command, a LIST and READ
program 26-2, a MODIFY RDN program 26-3, a REMOVE
program 26-4, an ADD program 26-5 and a MODIFY program
26-6. Memory 26 further comprises an information area
5 26-7 for storing information read out from the DIB 14
and information input from the keyboard 6, and a work
area 26-8 used for the structure changing processing.

Explanation will be made of the standard
directory service operations through DUA 8 and DSA 12.

10 Figs. 4, 5 and 6 are views for explaining the
LIST operation, the READ operation and the ADD operation
which are the standard service operations. As seen from
these figures, the directory includes entry names and
the tree structure information indicative of correlation
15 among entries and the directory is stored in the DIB 14.
Several operations for commands is used from a directory.
user are executed using the DIB 14.

(1) LIST operation

In this LIST operation, all names of entries
20 directly lower than a designated entry are returned to
DUA 8. When it is desired to know names of lower
entries of an entry B, as shown in Fig. 4, a LIST
command (entry name = B) is issued from the DUA 8 to the
DSA 12. The LIST command is supplied to the DSA 12
25 through the DUA 8. In response to this LIST command,
the DSA 12 searches the DIB 14 to thereby know that
names of entries directly lower of the entry B are B1,
B2 and B3. These entry names are responded to the DUA

1 8. Likewise, when the LIST command is issued to the DSA
12 through the DUA 8 for each of the entries B1, B2 and
B3, entry names X and Y are responded for the entry B1
and an entry name Z is responded for the entry B2.
5 However, since the leaf-entry B3 has no directly lower
entry, "No entry" is responded for the LIST command. By
executing the LIST operation until all the leaf-entries
are responded, the directory user can know a structure
portion of the directory information tree (DIT)
10 structure.

(2) READ operation

In this operation, information of a designated
entry such as an attribute type, an attribute value, and
like is read out. The directory user issues a READ
15 command to the DSA 12 through the DUA 8. In response to
this command, information possessed by the designated
entry can be known. The entire or part of the
information can be obtained in accordance with a
parameter of the command. As seen from Fig. 5, the
20 information of each entry is composed of a relative
distinguished name for distinguishing the designated
entry from other entries and its attribute data which
includes an attribute type and attribute values.

(3) ADD Entry operation

25 In this operation, an entry is added at a
lower portion than a designated leaf-entry in the tree
structure. The directory user sets the name of an entry
to be added and also its attribute information (which is

1 not occasionally included) in a parameter to issue an
ADD entry command to the DSA 12 through the DUA 8. In
an example of Fig. 6, an entry name B.B4 and its
attribute information are set in the parameter. The
5 entry name B.B4 means an entry B4 which is lower than
the entry B and has the relative distinguished name B4.
In response to the ADD Entry command, the DSA 12
searches the DIB 14 to check whether or not the entry
B.B4 is already present in the DIT structure stored in
10 the DIB 14. If the entry B.B4 is not present there,
this entry is added at a designated position of the DIT
structure. If the entry B.B4 is already present, a
response of error is sent to the DUA 8.

In addition to the above standard service
15 commands, operations for a REMOVE Entry command for
removing a designated entry, a MODIFY command for
modifying attribute information and an MODIFY RDN for
modifying the inherent name of a designated entry, and
like can be provided. It should be noted that the
20 REMOVE command and the MODIFY RDN command of these
commands can be issued only for leaf-entries.

Referring to Fig. 2, an operation of the first
embodiment of the present invention will be explained in
connection with processing for modifying the inherent
25 name of a non-leaf entry.

Now it is assumed that a directory user
intends to modify the inherent name of an object entry
of a DIT structure stored in the DIB 14. In this case,

1 generally, the user does not know whether the object
entry is a non-leaf entry or a leaf entry. Hence,
first, the user issues a standard command for modifying
the name of the object entry, MODIFY RDN to the decoding
5 section 22 from the keyboard 6. The decoding section
22, when the issued command is one of predetermined
standard commands, outputs that command to the DUA 8.
In response to this command, the DUA 8 requests, through
network 10, the DSA 12 to execute the command.

10 The DSA 12 examines the storage contents of
the DIB 14 to check if the entry designated by the
command is a leaf entry or not. If the designated entry
is a leaf entry, its name will be modified. If the
designated entry is a non-leaf entry, on the basis of
15 the response from the DSA 12, the DUA 8 transfers, to
the display unit 4 through the interface unit 2, a
response that the name of the designated entry, which is
a non-leaf entry, cannot be modified.

If regardless of this response, the directory
20 user still desires to modify the name of the designated
non-leaf entry, the user issues to the decoding section
22 a name modification command with an old inherent name
and a new inherent name as a parameter. The decoding
section 22 decodes this command to activate the
25 structure changing section 24. In accordance with main
program 26-1, and programs 26-2 to 26-6 for issuing
predetermined directory standard commands, the section
24 executes the structure changing processing, e.g. the

1 name modification processing in this embodiment. In
this case, the information area 26-7 in the memory 26 is
used to temporarily store information of the designated
entry and of its lower entries. The DUA 8 receives
5 commands from the section 24 and requests the DSA 12 to
execute these commands.

The structure changing section 24 receives a
command for modifying the name of a designated non-leaf
entry and performs operations in accordance with the
10 flowchart of Fig. 2 to modify the name of the designated
non-leaf entry. The flowchart will be explained below.

In step 122, the decoding section 22 receives,
from a directory user, a name modification command with
as a parameter a current inherent name as an old
15 inherent name and a new inherent name. The old inherent
name of the object entry is "Japan - A corporation - B
division - a department - second section". The "second
section" is a relative distinguished name of the object
entry. The new inherent name has the same form with the
20 relative distinguished name of "third section".

In step 124, the section 24 is activated in
accordance with the program 26-1. In order to acquire
all the entries for which modification should be made,
the section 24 executes the program 26-2. As a result,
25 a LIST command is repeatedly issued to the DUA 8 to
obtain all the entries directly lower than a designated
non-leaf entry, the entries directly lower than each of
the lower entries thus obtained, and so on. Thus, the

1 DIT structure information of the designated entry and
its all lower entries can be obtained.

In step 126, the DIT structure information is
stored in the information area 26-7 of the memory 26 as
5 shown in Fig. 7, but attribute information is not stored
yet.

In step 128, READ commands are issued to the
DUA 8 on the basis of the program 202 to obtain the
attribute information. Then, data link as shown in Fig.
10 7 is created in the work area 26-8 in accordance with
the structure information stored in the information area
26-7. Each data in the data link includes a pointer and
a flag. Each flag is set to "1" when the data link is
created and it is reset to "0" whenever the READ command
15 for the entry is issued. Respective READ commands are
executed to obtain the attribute information of the
designated entry and its all lower entries.

In step 130, the attribute information thus
obtained is stored in the information area 26-7 of the
20 memory 26 as shown in Fig. 7.

In step 132, after all the flags are set
again, the REMOVE Entry commands are sequentially issued
to the DUA 8 in accordance with a data link reverse to
the data link as described above. In response to each
25 REMOVE Entry command received through the network 10,
the DSA 12 sequentially removes the designated entry,
"second section", and its all lower entries, "Tanaka,
Yamada and Aoki", from the DIT structure in the DIB 14.

1 In step 134, after all the flags are set again
in accordance with the decoded command, the ADD Entry
commands are sequentially issued to the DUA 8 in accord-
ance with the data link, the DIT structure information
5 and the attribute information of each entry on the basis
of the program 26-5. Then, an entry with the new
inherent name containing a relative distinguished name
"third section" is added. Further, the entries lower
than the object entry with the old inherent name are
10 added at a lower portion than the added entry, with new
inherent names. In this way, the entry with the new
inherent name, and its all lower entries are added to
the DIT structure.

 In step 136, the structure changing process is
15 ended, and completion thereof is informed to the
directory user through the display unit 4.

 According to this embodiment of the present
invention, in order to modify the name of a non-leaf
entry, only one command has been executed to change the
20 object entry and all the lower entries with the old
inherent names into those with the new inherent names.

 The manner of modifying the entry name as
described above is shown in Fig. 3 in terms of the DIT
structures before and after execution of the modifica-
25 tion. The example of Fig. 3 shows that the entry with
the name of "A corporation - B division - a department -
second section" is shifted to the position of the entry
with the name of "A corporation - B division - a depart-

1 ment - third section", and the lower entries in the
former directory tree structure such as Tanaka, Yamada
and Aoki who are members of the section, are also
shifted to corresponding positions in the latter
5 directory tree structure.

Incidentally, although in this embodiment, the
standard service commands for modifying an inherent name
of a non-leaf entry were used in the order of LIST,
READ, REMOVE Entry and ADD Entry, it is apparent that
10 the order of REMOVE Entry and ADD Entry may be reversed.

Further, the removal of a non-leaf entry can
be attained through the LIST operations and the REMOVE
Entry operations, and the addition of a non-leaf entry
can be attained through the LIST operations, the READ
15 operations and the ADD Entry operation.

Now referring to the flowchart of Fig. 8 and
Fig. 1, a second embodiment of the present invention
will be explained only in its differences from the first
embodiment.

20 In step 122, a message input from the keyboard
6 is received by the decoding section 22.

In step 142, the decoding section 22 decides
if the received message is one of predetermined standard
operation commands.

25 If the answer in step 142 is 'Yes', in step
146, the message is passed to the DSA 12 through the DUA
8 and the network 10. If the answer in step 142 is
'No', in step 144, the decoding section 22 decides if

1 the received message is one of specially defined
commands or an error message for the command sent to the
DSA 12 in the step 122.

If the answer in step 144 is 'Yes', the
5 operations of step 124 et seq. are executed as in the
same manner as in the first embodiment.

As described above, in accordance with the
present invention, a command for changing the directory
structure has only to be issued once to shift a non-leaf
10 entry and its all lower entries to another positions
with new inherent names.

CLAIMS:

1. A method for changing a tree structure of an OSI directory comprising the steps of:

decoding a command for a non-leaf entry in the directory tree structure;

sequentially issuing OSI directory standard commands for the non-leaf entry and its lower entries in the directory tree structure in accordance with the decoding result; and

executing operations respectively corresponding to the issued standard commands.

2. A method according to claim 1, wherein

the command is a modification command for changing a first non-leaf entry with a first inherent name to a second non-leaf entry with a second inherent name, and

said command issuing step comprises the steps of:

issuing LIST commands to obtain a first structure portion including the first non-leaf entry and its lower entries;

issuing READ commands, in accordance with the first structure portion, to obtain attribute information of each entry of the first structure portion;

issuing REMOVE Entry commands, in accordance with the first structure portion, to remove each entry of the first structure portion from the directory tree structure; and

issuing ADD Entry commands, in accordance with the first structure portion, the second inherent name and the attribute information, to generate a second structure portion, which includes the second non-leaf entry with the second inherent name and the lower entries.

3. A method according to claim 1, wherein the command is a removal command for removing a first non-leaf entry with a first inherent name, and

said command issuing step comprises the steps of:

issuing LIST commands to obtain a first structure portion including the first non-leaf entry and its lower entries; and

issuing REMOVE Entry commands, in accordance with the first structure portion, to remove each entry of the first structure portion from the directory tree structure.

4. A method according to claim 1, wherein the command is an insert command for inserting a first non-leaf entry with a first inherent name under a designated entry of the directory tree structure, and

said command issuing step comprises the steps of:

issuing LIST commands to obtain a first structure portion including lower entries than the designated entry;

issuing READ commands, in accordance with the

first structure portion, to obtain attribute information of each entry of the first structure portion;

issuing REMOVE Entry commands in accordance with the first structure portion, to remove each entry of the first structure portion from the directory tree structure; and

issuing ADD Entry commands, in accordance with the first structure portion, the first non-leaf entry with the first inherent name and the obtained attribute information, to add the first non-leaf entry and the lower entries than the designated entry.

5. A method for changing a tree structure of an OSI directory, comprising the steps of:

performing by a directory service agent (DSA), an operation corresponding to a DSA message when the DSA message is one of OSI directory standard commands;

generating an error message by the DSA when an error occurs during the operation;

decoding a message;

supplying the message to the DSA as the DSA message when it is determined from the decoded result that the message is one of the standards commands; and

issuing to said DSA each of predetermined standard commands as the DSA message when it is determined from the decoded result that the message is the error message.

6. A method according to claim 5, wherein the command is a modification command for

shifting a first non-leaf entry with a first inherent name to a second non-leaf entry with a second inherent name, and

said command issuing step comprises the steps of:

issuing LIST commands to obtain a first structure portion including the first non-leaf entry and its lower entries;

issuing READ commands, in accordance with the first structure portion, to obtain attribute information of each entry of the first entry structure portion;

issuing REMOVE Entry commands, in accordance with the first structure portion, to remove each entry of the first structure portion; and

issuing ADD Entry commands, in accordance with the first structure portion, the second inherent name and the attribute information, to generate a second structure portion, which includes the second non-leaf entry with the second inherent name and lower entries corresponding to the first structure portion lower entries.

7. A method according to claim 5, wherein the command is a removal command for removing a first non-leaf entry with a first inherent name, and

said command issuing step comprises the steps of:

issuing LIST commands to obtain a first structure portion including the first non-leaf entry and

its lower entries; and

issuing REMOVE Entry commands, in accordance with the first structure portion, to remove each entry of the first structure portion from the directory tree structure.

8. A method according to claim 5, wherein the command is an insert command for inserting a first non-leaf entry with a first inherent name under a designated entry of the directory tree structure, and

said command issuing step comprises the steps of:

issuing LIST commands to obtain a first structure portion including lower entries than the designated entry;

issuing READ commands, in accordance with the first structure portion, to obtain attribute information of each entry of the first structure portion;

issuing Remove Entry commands in accordance with the first structure portion, to remove each entry of the first structure portion from the directory tree structure; and

issuing ADD Entry commands, in accordance with the first structure portion, the first non-leaf entry with the first inherent name and the attribute information, to add the first non-leaf entry and the lower entries than the designated entry.

9. A method according to claim 5, further comprising the step of, when the message is one of

specially defined commands, sequentially issuing OSI directory standard commands as the DSA message to said DSA in accordance with the message.

10. A directory service system for changing a tree structure of OSI director information, comprising:

directory information storage means for storing the OSI directory information;

directory service agent (DSA) means responsive to an input standard command for issuing an error message when the standard command is directed to a non-leaf entry of the directory information tree structure stored in said directory information storage means, and executing an operation corresponding to the standard command when the standard command is directed to a leaf entry thereof;

decoding means for decoding an input message, outputting the message to said DSA means when the message is the standard command and issuing a command issuing instruction when the message is the error message from said DSA means or one of predetermined commands; and

interface means responsive to the command issuing instruction, for sequentially issuing to said DSA means standard commands selected in accordance with the message.

11. A directory service system according to claim 10, wherein the message is a modification command for shifting a first non-leaf entry with a first inherent

name in the directory information tree structure to a second non-leaf entry with a second inherent name and one of the predetermined commands, or the message is a MODIFY RDN command for shifting the first non-leaf entry with the first inherent name in the directory information tree structure to the second non-leaf entry with the second inherent name as the standard command; and

said interface means comprises:

means for issuing LIST commands to obtain a first structure portion including the first non-leaf entry and its lower entries;

means for issuing READ commands, in accordance with the first structure portion, to obtain attribute information of each entry of the first structure portion;

means for issuing REMOVE Entry commands, in accordance with the first structure portion, to remove each entry from the first structure portion; and

means for issuing ADD Entry commands, in accordance with the first structure portion, the second inherent name and the attribute information, to generate a second structure portion including the second non-leaf entry with the second inherent name and lower entries corresponding to the first structure portion lower entries.

12. A directory service system according to claim 10, wherein the message is a removal command for removing a first non-leaf entry with a first inherent name

from the directory information tree structure and one of the predetermined commands or a REMOVE Entry command, and

said interface means comprises:

means for issuing LIST commands to obtain a first structure portion including the first non-leaf entry and its lower entries; and

means for issuing REMOVE Entry commands, in accordance with the first structure portion, to remove each entry of the first structure portion from the directory information tree structure.

13. A directory service system according to claim 10, wherein the message is an insert command for inserting a first non-leaf entry with a first inherent name into the directory information tree structure under a designated entry and one of the predetermined commands or an ADD Entry command, and

said interface means comprises:

means for issuing LIST commands to obtain a first structure portion including lower entries than the designated entry;

means for issuing READ commands in accordance with the first structure portion to obtain attribute information of each entry of the first structure portion;

means for issuing Remove Entry commands in accordance with the first structure portion, to remove the first structure portion from the directory

information tree structure; and

means for issuing ADD Entry commands in accordance with the first structure portion, said second inherent name, and the attribute information, to add the first non-leaf entry and the lower entries.

14. A directory service system substantially as herein described with reference to the accompanying drawings.